Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
Fakulteta za matematiko in fiziko

Blaž Oven

Računanje realnih ničel funkcij z uporabo Čebiševih polinomov

diplomsko delo
univerzitetni študijski program prve stopnje
računalništvo in matematika

Mentorica: izr. prof. Marjeta Krajnc

Ljubljana 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko in Fakulteta za matematiko in fiziko izdajata naslednjo nalogo:

Tematika naloge:

Robustne algoritme za računanje realnih ničel zvezne funkcije na nekem intervalu lahko dobimo tako, da funkcijo aproksimiramo s Čebiševim polinomom in izračunamo njegove ničle. V diplomskem delu na kratko predstavite Čebiševe polinome in njihove lastnosti. Predstavite algoritma za računanje ničel polinoma zapisanega v Čebiševi bazi, ki sta opisana v [1]. Prvi temelji na pridruženi matriki, drugi pa na postopku delitve (subdivizije) intervala.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Blaž Oven sem avtor diplomskega dela z naslovom:

Računanje realnih ničel funkcij z uporabo Čebiševih polinomov (angl. *Calculating real roots using Chebyshev polynomials*)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. Marjetke Krajnc,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 17. januarja 2015 Podpis avtorja:

Zahvaljujem se bližnjim za podporo pri končanju študija ter kolegu iz IŠRM za vso pomoč in nasvete. Hvala tudi mentorici, izr. prof. Marjetki Krajnc, za pomoč pri izbiri teme diplomskega dela in koordinaciji ter za ves trud in čas, ki si ga je vzela za nastanek tega dela. Hvala lepa za potrpežljivost, nasvete in vse popravke.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
CTP	convert to powers	metoda pretvorbe v potence
DD	degree-doubling	metoda podvajanja stopenj
FFT	fast Fourier transform	hitra Fourierova transformacija
DCT	discrete cosine transform	diskretna kosinusna transformacija

Povzetek

V diplomskem delu bomo predstavili Čebiševe polinome prve in druge vrste, njihove lastnosti ter Čebiševo vrsto. Uporabili bomo Čebiševe polinome prve vrste za iskanje ničel gladke funkcije f na danem intervalu. Najprej bomo funkcijo aproksimirali z Čebiševimi polinomi in nato nad končno Čebiševo vrsto uporabili polinomske iskalnike ničel. V nadaljevanju pa bomo predstavili, kako najti ničle polinomske funkcije na nekem intervalu, ki ga bomo pri nekaterih algoritmih razdelili na podintervale z namenom natančnejšega in tudi hitrejšega iskanja ničel. Predstavljenih bo nekaj algoritmov in njihova uporaba, pa tudi njihove zahtevnosti, slabosti in omejitve. V okviru dela smo algoritme tudi sprogramirali v programu Matlab. Njihova praktična uporaba bo predstavljena na primerih.

Ključne besede: ničle, Čebiševi polinomi, Čebiševa vrsta, pretvorba v potence, podvajanje stopnje.

Abstract

In this work, Chebyshev polynomials of the first and the second kind, their properties and the Chebyshev series will be examined. We will use Chebyshev polynomials of the first kind to find roots of the smooth function f on the given interval. At first the function will be approximated and then the polynomial root-finders on the truncated Chebyshev series will be used. In the next chapter we will study how to find roots of a polynomial function on the interval which we will, with some algorithms, divide on subintervals with the purpose of more accurate and faster finding of the roots. Different algorithms and their use, their complexity and their strengths and weaknesses will be presented. During this work we have also programmed these algorithms in Matlab. We will show their practical application on some examples.

Keywords: roots, Chebyshev polynomials, Chebyshev series, Convert to powers, Degree doubling.

Poglavje 1

Uvod

Za iskanje ničel podane funkcije obstaja zelo veliko algoritmov. Nekateri so dobri, spet drugi imajo pomanjkljivosti. V tem delu smo se osredotočili na uporabo Čebiševih polinomov kot sredstva za iskanje ničel. V prvem delu najprej predstavimo Čebiševe polinome prve in druge vrste ter povezavo med njimi in njihove lastnosti, kot so kompozitum in produkt, ničle in ekstremi, odvajanje in integracija, ortogonalnost ter norma. V naslednjem poglavju pa najprej prikažemo iskanje ničel gladke funkcije f. Predstavimo, kako izračunamo Čebiševe koeficiente, si v splošnem pogledamo delitev intervalov in problem skaliranja funkcije. Predstavimo tudi, kaj se zgodi v primeru negladke funkcije in si pogledamo, kakšen je postopek, da pridemo od končne Čebiševe vrste do polinoma v standardni bazi. Spoznamo tudi metodo podvajanja stopnje in kako poteka iskanje ničel na celotni realni osi. V tem drugem delu diplomskega dela pa se tudi naučimo, kako gladko funkcijo f aproksimiramo, zapišemo v obliki končne Čebiševe vrste in le-to nato vstavimo v polinomski algoritem za iskanje ničel. V tretjem delu pa se seznanimo z algoritmi, s katerimi poiščemo ničle polinomske funkcije f. Ukvarjamo se z algoritmi, ki razdelijo celotni interval, na katerem iščemo ničle, na manjše podintervale. Nato polinom aproksimiramo na vsakem podintervalu posebej in iščemo ničle na posameznem podintervalu. Najprej predstavimo metodo računanja ničel preko Čebišev-Frobeniusove matrike, za katero potrebujemo

le Čebiševe koeficiente aproksimacijske funkcije. Za tem pa prikažemo delitev intervalov, kjer spoznamo algoritem Megakubi, 13-N in 10-N. Predstavimo tudi možne pohitritve iskanja ničel z določanjem brezničelnih intervalov ter napake teh metod. V zadnjem poglavju predstavimo uporabljene sprogramirane metode, ki so priložene delu in so bile predstavljene v predhodnjih poglavjih. Med seboj jih s pomočjo primera primerjamo po natančnosti in hitrosti računanja.

Poglavje 2

Čebiševi polinomi

2.1 Čebiševi polinomi

Čebiševi polinomi oz. polinomi Čebiševa so v matematiki zaporedje ortogonalnih polinomov, ki so povezani z de Moivreovo formulo in jih lahko preprosto določimo rekurzivno kot na primer Fibonaccijeva ali Lucasova števila. Imenujejo se po Pafnutiju Lvoviču Čebiševu. Poznamo dve vrsti Čebiševih polinomov, in sicer polinome Čebiševa prve vrste, označene s T_n, in polinome Čebiševa druge vrste, označene z U_n. Črka T se uporablja zaradi različnih prečrkovanj priimka Čebišev: Tchebyshef ali Tschebyscheff. Polinomi Čebiševa T_n ali U_n so polinomi stopnje n.

Polinomi Čebiševa prve vrste so pomembni v teoriji aproksimacije, saj se njihove ničle, imenovane vozli Čebiševa, uporabljajo kot vozli pri polinomski interpolaciji. Ustrezna interpolacija zmanjša problem Rungejevega pojava in omogoča aproksimacijo, ki je blizu polinomu najboljše aproksimacije za zvezno funkcijo glede na maksimalno normo ([4]). Ta aproksimacija vodi neposredno k metodi numerične integracije imenovane Clenshaw-Curtisova kvadratura.

Polinomi Čebiševa prve in druge vrste so rešitve diferencialnih enačb Čebiševa

$$(1 - x^2)\frac{d^2y}{dx^2} - x\frac{dy}{dx} + n^2y = 0$$

in

$$(1-x^2)\frac{d^2y}{dx^2} - 3x\frac{dy}{dx} + n(n-2)y = 0.$$

2.1.1 Definicije Čebiševih polinomov

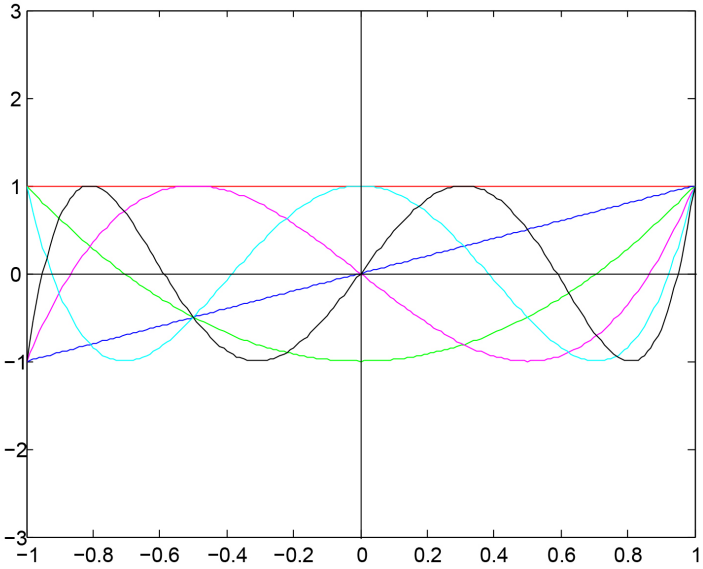
Čebiševe polinome prve vrste lahko definiramo preko rekurzivne zveze

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned} \tag{2.1}$$

Prvih pet Čebiševih polinomov prve vrste je enakih

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ T_5(x) &= 16x^5 - 20x^3 + 5x \end{aligned}$$

in so predstavljeni na sliki 2.1.



Slika 2.1: Grafični prikaz prvih petih Čebiševih polinomov, in sicer T_0 rdeča, T_1 modra, T_2 zelena, T_3 magenta, T_4 cyan in T_5 črna krivulja.

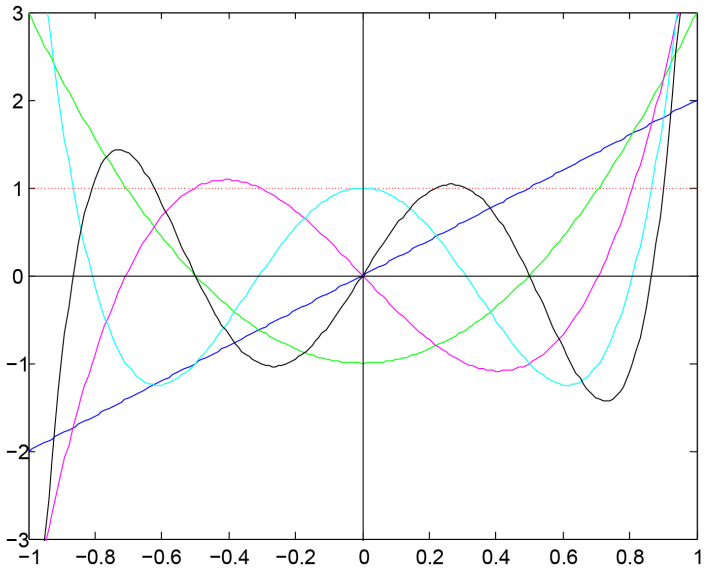
Čebiševi polinomi druge vrste pa so definirani z

$$\begin{aligned} U_0(x) &= 1, \\ U_1(x) &= 2x, \\ U_{n+1}(x) &= 2xU_n(x) - U_{n-1}(x). \end{aligned}$$

Prvih pet Čebiševih polinomov se glasi

$$\begin{aligned} U_0(x) &= 1, \\ U_1(x) &= 2x, \\ U_2(x) &= 4x^2 - 1, \\ U_3(x) &= 8x^3 - 4x, \\ U_4(x) &= 16x^4 - 12x^2 + 1, \\ U_5(x) &= 32x^5 - 32x^3 + 6x \end{aligned}$$

in so predstavljeni na sliki 2.2.



Slika 2.2: Grafični prikaz prvih petih Čebiševih polinomov, in sicer U_0 rdeča, U_1 modra, U_2 zelena, U_3 magenta, U_4 cyan in U_5 črna krivulja.

Ekvivalentno lahko definiramo Čebiševe polinome z uporabo trigonometričnih funkcij. Čebišev polinom prve vrste je na intervalu [-1,1] definiran kot

$$T_n(x) = \cos(n \arccos(x)) \tag{2.2}$$

oziroma

$$T_n(x) = \cos(n\vartheta), \quad x = \cos(\vartheta), \quad \vartheta \in [0, \pi], \tag{2.3}$$

za $n = 0, 1, 2, 3, \ldots$.
Tako dobimo

$$T_0(x) = \cos(0\vartheta) = 1 \quad \text{in} \quad T_1(x) = \cos(\vartheta) = x,$$

iz česar sledi

$$T_2(x) = \cos(2\vartheta) = 2\cos^2\vartheta - 1 = 2x^2 - 1$$

ter

$$T_3(x) = \cos(3\vartheta) = 2\cos\vartheta\cos(2\vartheta) - \cos(\vartheta) = 4\cos^3\vartheta - 3\cos\vartheta = 4x^3 - 3x.$$

Podobno velja za ostale Čebiševe polinome višjih stopenj.

Čebiševi polinomi druge vrste pa zadoščajo enakosti

$$U_n(x) = \frac{\sin((n+1)\vartheta)}{\sin\vartheta}, \quad x = \cos(\vartheta). \tag{2.4}$$

2.1.2 Lastnosti Čebiševih polinomov

Kompozitum in produkt Čebiševih polinomov

Prva lastnost, kompozitum dveh polinomov, se glasi

$$T_n(T_m(x)) = T_{nm}(x). \tag{2.5}$$

Naredimo izpeljavo formule (2.5). Za T_n in T_m uporabimo definicijo 2.2. Torej

$$T_n(x) = \cos(n \arccos(x))$$

in

$$T_m(x) = \cos(m \arccos(x)),$$

iz česar nato sledi

$$T_n(T_m(x)) = \cos(n \arccos(\cos(m \arccos(x)))) = \cos(nm \arccos(x)).$$

Ker se velikokrat srečujemo s produktom Čebiševih poliomov, si poglejmo, kako jih množimo med seboj. Produkt lahko zapišemo kot kombinacijo Čebiševih polinomov z manjšo in višjo stopnjo. Če predpostavimo $m \geq n$ in $n \neq 0$, za produkt Čebiševih polinomov prve vrste velja

$$2T_m(x)T_n(x) = T_{m+n}(x) + T_{m-n}(x). \tag{2.6}$$

Naveden zapis sledi iz enakosti

$$2\cos\alpha\cos\beta = \cos(\alpha+\beta) + \cos(\alpha-\beta)$$

za $\alpha = m \arccos x$ in $\beta = n \arccos x$. Če v (2.6) vstavimo $m = 1$, dobimo že znano rekurzivno formulo, pri izbiri $m = n+1$ pa dobimo rekurzivni formuli

za vse lihe in sode Čebiševe polinome, odvisne od paritete najnižjega m. Tako dobimo tri uporabne formule

$$\begin{aligned} T_{2n}(x) &= 2T_n^2(x) - T_0(x) = 2T_n^2(x) - 1, \\ T_{2n+1}(x) &= 2T_{n+1}(x)T_n(x) - T_1(x) = 2T_{n+1}(x)T_n(x) - x, \\ T_{2n-1}(x) &= 2T_{n-1}(x)T_n(x) - T_1(x) = 2T_{n-1}(x)T_n(x) - x. \end{aligned}$$

Produkt Čebiševih polinomov druge vrste pa zapišemo kot

$$U_m(x)U_n(x) = \sum_{k=0}^{n} U_{m-n+2k}(x). \tag{2.7}$$

Z uporabo definicije 2.4 potrdimo, da formula (2.7) res drži. Torej

$$\sin\vartheta \cdot \sum_{k=0}^{n} U_{m-n+2k}(\cos\vartheta) = \Im\left(\sum_{k=0}^{n} \mathrm{e}^{\mathrm{i}(m-n+2k+1)\vartheta}\right) = \Im(z),$$

kjer je

$$z = \mathrm{e}^{\mathrm{i}(m-n+1)\vartheta} \sum_{k=0}^{n} \mathrm{e}^{2k\mathrm{i}\vartheta} = \mathrm{e}^{\mathrm{i}(m-n+1)\vartheta} \frac{\mathrm{e}^{2(n+1)\mathrm{i}\vartheta} - 1}{\mathrm{e}^{2\mathrm{i}\vartheta} - 1} = \mathrm{e}^{\mathrm{i}(m+1)\vartheta} \frac{\sin((n+1)\vartheta)}{\sin\vartheta}.$$

Torej

$$\sum_{k=0}^{n} U_{m-n+2k}(\cos\vartheta) = \frac{\sin((m+1)\vartheta)}{\sin\vartheta} \cdot \frac{\sin((n+1)\vartheta)}{\sin\vartheta} = U_m(\cos\vartheta) \cdot U_n(\cos\vartheta).$$

Če, tako kot pri Čebiševih polinomih prve vrste, izberemo $n = 2$, dobimo rekurzivno formulo, ki je simetrična glede na najnižjo vrednost m ($m = 2$ ali $m = 3$)

$$U_{m+2}(x) = U_2(x)U_m(x) - U_m(x) - U_{m-2}(x) = U_m(x)(U_2(x) - 1) - U_{m-2}(x). \tag{2.8}$$

Povezava med Čebiševimi polinomi prve in druge vrste

Čebiševi polinomi prve in druge vrste so povezani med seboj. Prvi dve relaciji, ki jih zasledimo v literaturi [8], sta

$$\frac{d}{dx}T_n(x) = nU_{n-1}(x), \quad n = 1, \ldots$$

ter

$$T_n(x) = \frac{1}{2}(U_n(x) - U_{n-2}(x)). \tag{2.9}$$

Izpeljani sta iz relacije

$$2T_n(x) = \frac{1}{n+1}\frac{d}{dx}T_{n+1}(x) - \frac{1}{n-1}\frac{d}{dx}T_{n-1}(x), \quad n = 2, 3, \ldots.$$

Naslednji relaciji, ki jih navaja [8], sta

$$T_{n+1}(x) = xT_n(x) - (1-x^2)U_{n-1}(x) \tag{2.10}$$

in

$$T_n(x) = U_n(x) - xU_{n-1}(x).$$

Obe relaciji lahko izpeljemo iz trigonometričnih definicij (2.3) in (2.4). Pokažimo na primer (2.10) z uporabo (2.3) in $x = \cos\vartheta$

$$\begin{aligned} T_{n+1}(x) &= T_{n+1}(\cos(\vartheta)) \\ &= \cos((n+1)\vartheta) \\ &= \cos(n\vartheta)\cos(\vartheta) - \sin(n\vartheta)\sin(\vartheta) \\ &= T_n(\cos(\vartheta))\cos(\vartheta) - U_{n-1}(\cos(\vartheta))\sin^2(\vartheta) \\ &= xT_n(x) - (1-x^2)U_{n-1}(x). \end{aligned}$$

Povezavo med Čebiševimi polinomi prve in druge vrste lahko izrazimo iz (2.9) in rekurzivno nadaljujemo. Posledično dobimo

$$U_n(x) = 2\sum_{j=0}^{\frac{n}{2}} T_{2j}(x),$$

kjer je n sod, oziroma

$$U_n(x) = 2\sum_{j=1}^{\frac{n+1}{2}} T_{2j-1}(x) - 1,$$

kjer je n lih.

Ničle in ekstremi

Čebišev polinom stopnje n ima, ne glede na to ali je prve ali druge vrste, n različnih ničel, imenovanih Čebiševe ničle na intervalu $[-1, 1]$. Ničlam Čebiševega polinoma prve vrste pravimo tudi vozlišča, saj jih uporabljamo kot vozle pri polinomski interpolaciji ([8]). Če uporabimo trigonometrično definicijo in upoštevamo dejstvo, da je $\cos((2k+1)\frac{\pi}{2}) = 0$, lahko hitro izpeljemo, da so ničle Čebiševih polinomov T_n prve vrste enake

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right), k = 1, \ldots, n.$$

Podobno velja, da so ničle polinomov U_n druge vrste oblike

$$x_k = \cos\left(\frac{k}{n+1}\pi\right), k = 1, \ldots, n.$$

Ekstremi T_n na intervalu $[-1, 1]$ so doseženi v točkah

$$x_k = \cos\left(\frac{k}{n}\pi\right), k = 1, \ldots, n.$$

Posebna lastnost Čebiševih polinomov prve vrste je, da na tem intervalu vrednost extrema znaša 1 ali -1 ([8]).

Odvajanje in integracija

Z uporabo definicije Čebiševih polinomov v njihovi trigonometrični obliki lahko izpeljemo

$$\begin{aligned} \frac{dT_n}{dx} &= nU_{n-1}, \\ \frac{dU_n(x)}{dx} &= \frac{(n+1)T_{n+1}(x) - xU_n(x)}{x^2-1}, \\ \frac{d^2T_n(x)}{dx^2} &= n\frac{nT_n(x) - xU_{n-1}(x)}{x^2-1} = n\frac{(n+1)T_n(x) - U_n(x)}{x^2-1}. \end{aligned} \tag{2.11}$$

Pokažimo na primer izpeljavo $\frac{dT_n}{dx} = nU_{n-1}$. Ko odvajamo T_n v obliki (2.2) po x dobimo

$$\frac{dT_n(x)}{dx} =$$
$$\frac{d\cos(n\arccos(x))}{dx} =$$
$$= n\frac{\sin(n\arccos(x))}{\sqrt{1-x^2}} = n\frac{\sin(n\arccos(x))}{\sin(\arccos(x))} =$$
$$= nU_{n-1}(x).$$

Izpeljimo še formulo za odvod Čebiševih polinomov druge vrste. Odvajamo U_n v obliki (2.4) in dobimo:

$$\frac{dU_n(x)}{dx} =$$
$$\left(\frac{-\sqrt{1-x^2}\cos\left((n+1)\arccos(x)\right)(n+1)}{\sqrt{1-x^2}} + \frac{\sin\left((n+1)\arccos(x)\right)x}{\sqrt{1-x^2}}\right)\frac{1}{1-x^2}$$
$$= \frac{-T_{n+1}(x)(n+1)}{1-x^2} + \frac{xU_n(x)}{1-x^2}$$
$$= \frac{(n+1)T_{n+1}(x) - xU_n(x)}{x^2-1}.$$

Za izpeljavo zadnje formule v (2.11) uporabimo prvo in drugo formulo. Zadnji dve formuli sta lahko numerično problematični zaradi deljenja z 0 pri $x = 1$ in $x = -1$. Pri tem si pomagamo s sledečo lemo.

Lema 2.1 *Za Čebiševe polinome prve vrste velja*

$$\left.\frac{d^2T_n(x)}{dx^2}\right|_{x=1} = \frac{n^4-n^2}{3},$$
$$\left.\frac{d^2T_n(x)}{dx^2}\right|_{x=-1} = (-1)^n\frac{n^4-n^2}{3}.$$

Dokaz. Drugi odvod Čebiševega polinoma prve vrste je

$$T_n''(x) = n\frac{nT_n(x) - xU_{n-1}(x)}{x^2-1},$$

ki ob evaluaciji predstavlja problem pri $x = \pm 1$. Ker je funkcija polinomska, so tudi njeni odvodi polinomi. Želeno vrednost tako dobimo kot

$$T_n''(1) = \lim_{x\to 1} n\frac{nT_n(x) - xU_{n-1}(x)}{x^2-1}.$$

Razcepimo imenovalec

$$T_n''(1) = \lim_{x\to 1} n\frac{nT_n(x) - xU_{n-1}(x)}{(x-1)(x+1)} = \lim_{x\to 1} n\frac{\frac{nT_n(x)-xU_{n-1}(x)}{x-1}}{x+1}.$$

Ker mora obstajati tako limita kot celota, kot tudi limita števca in imenovalca, je

$$T_n''(1) = n\frac{\lim_{x\to 1}\frac{nT_n(x)-xU_{n-1}(x)}{x-1}}{\lim_{x\to 1}(x+1)} = \frac{n}{2}\lim_{x\to 1}\frac{nT_n(x) - xU_{n-1}(x)}{x-1}.$$

Imenovalec še vedno limitira k 0, kar implicira na to, da mora tudi števec limitirati k 0; torej $U_{n-1}(1) = nT_n(1) = n$. Ker oba, števec in imenovalec, limitirata k 0, lahko uporabimo L'Hospitalovo pravilo:

$$\begin{aligned}
T_n''(1) &= \frac{n}{2}\lim_{x\to 1}\frac{\frac{d}{dx}(nT_n(x) - xU_{n-1}(x))}{\frac{d}{dx}(x-1)} \\
&= \frac{n}{2}\lim_{x\to 1}\frac{d}{dx}\big(nT_n(x) - xU_{n-1}(x)\big) \\
&= \frac{n}{2}\lim_{x\to 1}\Big(n^2U_{n-1}(x) - U_{n-1}(x) - x\frac{d}{dx}\big(U_{n-1}(x)\big)\Big) \\
&= \frac{n}{2}\Big(n^2U_{n-1}(1) - U_{n-1}(1) - \lim_{x\to 1} x\frac{d}{dx}\big(U_{n-1}(x)\big)\Big) \\
&= \frac{n^4}{2} - \frac{n^2}{2} - \frac{1}{2}\lim_{x\to 1}\frac{d}{dx}\big(nU_{n-1}(x)\big) \\
&= \frac{n^4}{2} - \frac{n^2}{2} - \frac{T_n''(1)}{2},
\end{aligned}$$

od koder dobimo

$$T_n''(1) = \frac{n^4 - n^2}{3}.$$

Za $x = -1$ je dokaz podoben, pri čemer uporabimo enakost $T_n(-1) = (-1)^n$. □

V splošnem velja

$$\frac{d^pT_n(x)}{dx^p}\Big|_{x=\pm 1} = (\pm 1)^{n+p}\prod_{k=0}^{p-1}\frac{n^2-k^2}{2k+1}.$$

Če pogledamo prvi odvod $\frac{dT_n}{dx} = nU_{n-1}$, vidimo, da velja

$$\int U_n(x)dx = \frac{T_{n+1}(x)}{n+1} + C$$

in tako iz rekurzivnega izraza (2.1) za Čebiševe polinome prve vrste z upoštevanjem odvodov dobimo

$$\int T_n(x)dx = \frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1}\right) + C = \frac{nT_{n+1}(x)}{n^2-1} - \frac{xT_n(x)}{n-1} + C.$$

Ortogonalnost

Lastnost Čebiševih polinomov prve vrste je, da so ortogonalni glede na skalarni produkt

$$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)\frac{1}{\sqrt{1-x^2}}.$$

Podobno so Čebiševi polinomi druge vrste ortogonalni glede na

$$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)\sqrt{1-x^2}$$

in sicer velja

$$\int_{-1}^{1} T_n(x)T_m(x)\frac{dx}{\sqrt{1-x^2}} = \begin{cases} 0, & n \neq m \\ \pi, & n = m = 0 \\ \pi/2, & n = m \neq 0 \end{cases}.$$

Navedeno lahko dokažemo z uporabo $x = \cos(\theta)$ in definicije (2.3). Podobno lahko ugotovimo tudi, da na intervalu $[-1, 1]$ za Čebiševe polinome druge vrste z uporabo uteži $\sqrt{1-x^2}$ dobimo

$$\int_{-1}^{1} U_n(x)U_m(x)\sqrt{1-x^2}dx = \begin{cases} 0, & n \neq m \\ \pi/2, & n = m \end{cases}.$$

Pokažemo lahko, da so Čebiševi polinomi prve in druge vrste ortogonalni glede na določene diskretne skalarne produkte oblike

$$\langle f, g\rangle = \sum_{k=0}^{N-1} f(x_k)g(x_k)\rho(x_k),$$

kjer je $\rho(x_k) > 0$ utež in so $x_k = \cos\left(\frac{2k+1}{2N}\pi\right)$ ničle polinoma T_N. Pri tem velja

$$\sum_{k=0}^{N-1} T_i(x_k)T_j(x_k) = \begin{cases} 0, & i \neq j \\ N, & i = j = 0 \\ N/2, & i = j \neq 0 \end{cases}.$$

Podobno za polinome druge vrste z utežjo $\rho(x) = 1 - x^2$ velja

$$\sum_{k=0}^{N-1} U_i(x_k)U_j(x_k)(1-x_k^2) = \begin{cases} 0, & i \neq j \\ N/2, & i = j \end{cases}$$

in brez uporabe uteži

$$\sum_{k=0}^{N-1} U_i(x_k)U_j(x_k) = \begin{cases} 0, & parnost(i) \neq parnost(j) \\ N + N \cdot \min(i,j), & parnost(i) = parnost(j) \end{cases},$$

kjer $parnost(i)$ predstavlja sodost oziroma lihost. Če pa uporabimo N ničel Čebiševih polinomov druge vrste, to je $x_k = \cos\left(\pi\frac{k+1}{N+1}\right)$, pa dobimo

$$\sum_{k=0}^{N-1} U_i(x_k)U_j(x_k)(1-x_k^2) = \begin{cases} 0, & i \neq j \\ \frac{N+1}{2}, & i = j \end{cases}$$

in podobno brez uporabe uteži

$$\sum_{k=0}^{N-1} U_i(x_k)U_j(x_k) = \begin{cases} 0, & parnost(i) \neq parnost(j) \\ (\min(i,j)+1)(N - \max(i,j)), & parnost(i) = parnost(j) \end{cases}.$$

Za izpeljavo glej [8].

Minimalna ∞-norma

Izrek 2.1 *Izmed vseh polinomov stopnje $n \geq 1$ z vodilnim koeficientom enakim 1, je*

$$p(x) = \frac{1}{2^{n-1}} T_n(x)$$

polinom, ki ima na intervalu $[-1,1]$ *minimalno neskončno normo, to je*

$$\min_{q,stopnja\ q\leq n} ||q||_{\infty,[-1,1]} = \min_{q,stopnja\ q\leq n} \max_{x\in[-1,1]} |q(x)| = \left\| \frac{1}{2^{n-1}} T_n \right\|_{\infty,[-1,1]}.$$

Ta neskončna norma je enaka 2^{1-n} *in jo* $|p|$ *doseže v natanko* $n+1$ *točkah* $x = \cos\frac{k\pi}{n}, 0 \leq k \leq n$.

Dokaz. Predpostavimo, da imamo polinom w_n stopnje n z vodilnim koeficientom 1 in z neskončno normo na intervalu $[-1,1]$ manjšo od $\frac{1}{2^{n-1}}$. Definirajmo $f_n(x) = \frac{1}{2^{n-1}} T_n(x) - w_n(x)$, ki je polinom stopnje $\leq n-1$. Zaradi točk x, v katerih T_n doseže maksimum, imamo

$$|w_n(x)| < \left| \frac{1}{2^{n-1}} T_n(x) \right|.$$

Od tod sledi, da je

$$f_n(x) > 0 \text{ za } x = \cos\frac{2k\pi}{n}, \quad k = 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor,$$
$$f_n(x) < 0 \text{ za } x = \cos\frac{(2k+1)\pi}{n}, \quad k = 0, 1, \ldots, \left\lfloor \frac{n-1}{2} \right\rfloor.$$

Iz izreka, da zvezna funkcija $f : \mathbb{R} \to \mathbb{R}$ na zaprtem intervalu $[a,b]$ zavzame vsako vrednost med najmanjšo in največjo oz. iz njegove posledice, ki pravi, da če je f v krajiščih tega intervala nasprotno predznačena, ima na odprtem intervalu (a,b) f vsaj eno ničlo, sledi, da ima f_n vsaj n ničel ([3]). Ker pa je f_n polinom stopnje $n-1$, je to v nasprotju z osnovnim izrekom algebre, ki pravi, da ima lahko največ $n-1$ ničel. Torej smo prišli do protislovja in s tem zaključili dokaz. □

2.2 Čebiševa vrsta

Definirajmo najprej Čebiševe točke druge vrste, to so točke na intervalu [-1,1], definirane z

$$x_j = -\cos\left(\frac{j\pi}{N}\right), \quad 0 \leq j \leq N, \tag{2.12}$$

kjer je $N \geq 1$. V primeru $N = 0$ je $x_0 = 0$. Iz polinomske interpolacije vemo, da ne glede na vrednosti f_j, $j = 0, 1, \ldots, N$, obstaja natanko en polinomski interpolant p stopnje $\leq N$, za katerega velja $p(x_j) = f_j$, $j = 0, 1, \ldots, N$. Pravimo mu tudi Čebišev interpolant. Če imamo podatke definirane kot $f_j = (-1)^{N-j}$, $j = 0, 1, \ldots, N$, potem je $p(x) = T_N(x)$ Čebišev polinom stopnje N, kar je razvidno iz formule $T_N(x) = \cos(N \arccos(x))$.

Čebiševo vrsto zapišemo kot

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x), \tag{2.13}$$

kjer so a_k t.i. Čebiševi koeficienti, ki so definirani kot

$$a_k = \frac{1}{\pi} \int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}} dx, \quad k = 1,$$

$$a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}} dx, \quad k \geq 1.$$

Čebiševa vrsta obstaja za vsako funkcijo, ki je dovolj gladka oz. vsaj Lipschitzevo zvezna. Ta pogoj pomeni, da za vsak $x, y \in [a, b]$ obstaja $K \geq 0, K \in \mathbb{R}$, da velja $|f(x) - f(y)| \leq K|x - y|$, povzeto po [6]. Takšna vrsta konvergira enakomerno in absolutno na celotnem intervalu.

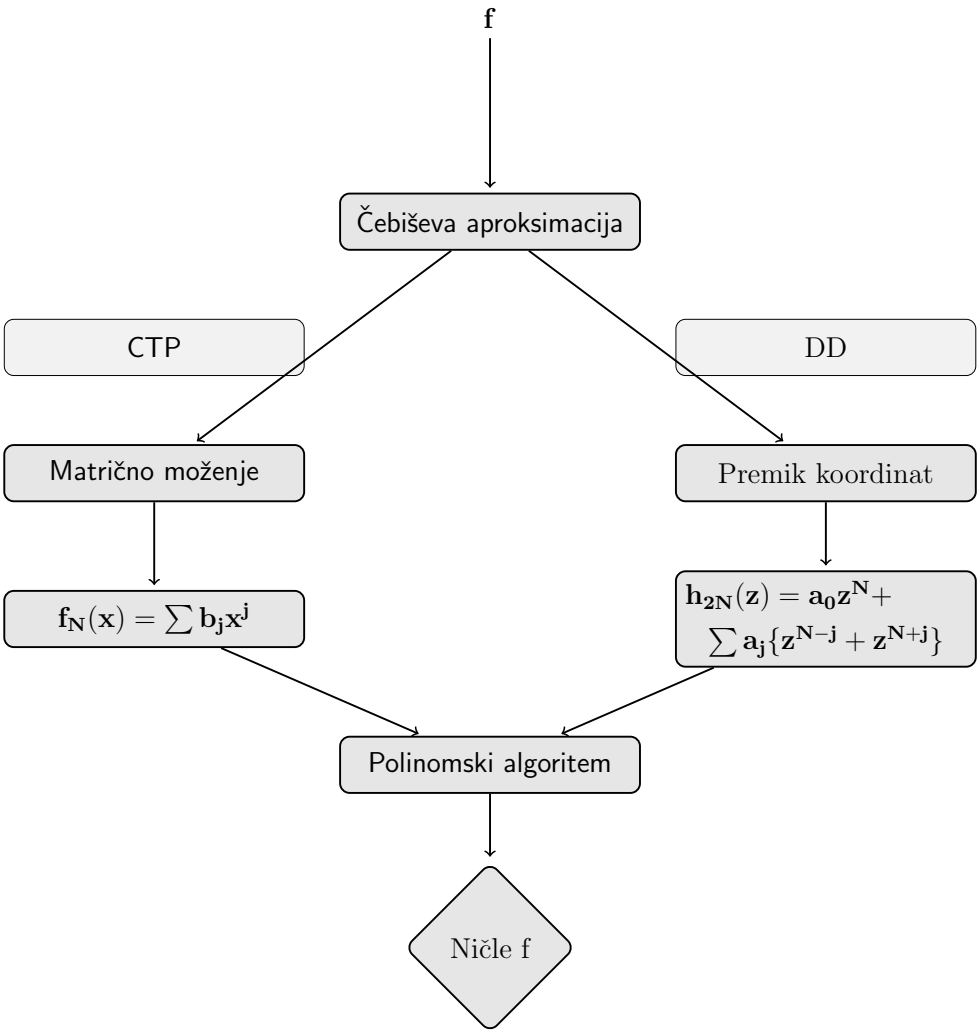
Poglavje 3

Iskanje ničel

3.1 Iskanje ničel preko standardnih algoritmov za ničle polinomov

Po podrobnejšem opisu Čebiševih polinomov in nekaj njihovih lastnosti v predhodnem poglavju, se v tem poglavju osredotočimo na problem iskanja ničel funkcij ene spremenljivke. Predstavimo, kako lahko za iskanje ničel nepolinomske funkcije na realnem intervalu uporabimo robustne polinomske algoritme. Kot prvi korak aproksimiramo f z zaporedjem Čebiševih polinomov, ki jih nato v drugem koraku preoblikujemo v končno vrsto oblike (2.13) in nato nad njo v tretjem koraku uporabimo enega izmed polinomskih algoritmov. Pri tem so kompleksne ničle in realne ničle zunaj izbranega intervala zanemarjene. Pri prvem koraku je najbolj učinkovito, če uporabimo Čebiševo interpolacijo na $(N+1)$ točkah, pri čemer je N enak potenci števila 2 in kjer Čebiševa vrsta vsebuje N členov. Vse že izračunane evaluacije funkcije f lahko znova uporabimo, ko stopnjo N povečamo. S postopkom prenehamo, ko je N dovolj velik in ko njegovo povečevanje ne prinese velikih sprememb v interpolantu. Opisali bomo dve metodi. Prva metoda, pretvorba v potence (ang. *convert to powers*, v nadaljevanju CTP), uporablja množenje malo slabše pogojenih matrik z namenom ustvarjanja polinoma stopnje N. Druga metoda, podvajanje stopenj (ang. *degree-doubling*, v nadaljevanju DD), defi-

nira polinom višje stopnje $2N$, ampak je veliko bolje pogojena. Prva metoda, ki je sicer hitrejša, pa nam omeji N na zmerne vrednosti, kar pa lahko vedno dosežemo z delitvijo intervala. Obe metodi omogočata simultano aproksimacijo večih ničel nepolinomske funkcije f na intervalu ne glede na to, ali so ničle enostavne ali večkratne. Osnovna ideja obeh metod je prikazana na sliki 3.1.



Slika 3.1: Prikaz osnovne ideje algoritmov.

Prvo vprašanje, ki se nam zastavi je, kako izračunati končno Čebiševo

vrsto, ki aproksimira dano funkcijo f na intervalu $[a, b]$. Uporabili bomo polinomsko interpolacijo. Najprej moramo imeti f evaluirano na množici diskretnih točk na izbranem intervalu. Čebiševe koeficiente nato izračunamo z množenjem matrike z vektorjem, kjer vektor vsebuje množico vrednosti funkcije f v izbranih točkah iz intervala $[a, b]$, elementi matrike pa so trigonometrične funkcije. Bolj natančno, interpolacijske točke za prvi korak izberemo po formuli

$$x_k = \frac{b-a}{2} \cos\left(\pi \frac{k}{N}\right) + \frac{b+a}{2}, \quad k = 0, 1, \ldots, N. \tag{3.1}$$

Izračunati moramo še vrednosti funkcije f v vsaki od interpolacijskih točk, to je

$$f_k = f(x_k), \quad k = 0, 1, \ldots, N.$$

Za drugi korak moramo izračunati elemente $(N+1) \times (N+1)$ interpolacijske matrike $Q = (q_{ij})_{i,j=0}^{N}$. Definiramo $p_0 = p_N = 2$ in $p_j = 1$, $j = 1, 2, \ldots, N-1$. Elementi interpolacijske matrike so nato po [2] enaki

$$q_{jk} = \frac{2}{p_j p_k N} \cos\left(j\pi \frac{k}{N}\right).$$

V zadnjem koraku izračunamo koeficiente, kar opravimo preko množenja matrike z vektorjem vrednosti, to je

$$(a_j)_{j=0}^{N} = Q\,(f_j)_{j=0}^{N}.$$

Interval $[a, b]$ preslikamo na $[-1, 1]$ z uporabo translacije

$$y = y(x) = \frac{2x - (b+a)}{b-a}, \quad x \in [a, b]. \tag{3.2}$$

Iskana aproksimacija je nato

$$f(y) \approx \sum_{j=0}^{N} a_j T_j(y) = \sum_{j=0}^{N} a_j \cos(j \arccos(y)).$$

Vprašanje je, pri katerem N dobimo dovolj dobro aproksimacijo. Najbolj sistematično je, da število točk podvajamo, dokler se aproksimacija izboljšuje in

spreminja. Vse prejšnje že izračunane vrednosti seveda ponovno uporabimo in s tem prihranimo čas. Zaustavitveni kriterij bomo predstavili v nadaljevanju.

Tretje vprašanje, ki se pojavi je, kako iz Čebiševe vrste učinkovito izračunati polinom v standardni bazi. V nadaljevanju si bomo pogledali dve metodi. Pri prvi metodi, to je CTP, so koeficienti pri potencah x-a produkt zgornje trikotne matrike z vektorjem Čebiševih koeficientov, elementi matrike pa so števila, ki so dobljena z rekurzivno enačbo. Drugi algoritem, po metodi DD, pa definira polinom, čigar stopnja je dvakratnik stopnje končne Čebiševe vrste. Realni deli ničel tega polinoma, ki ležijo na enotski krožnici v kompleksni ravnini, so ničle funkcije f na realnem intervalu $[a, b]$. Pri četrtem vprašanju nas zanima, kako stabilna je metoda CTP, saj je znano, da so ničle polinoma občutljive na majhne spremembe koeficientov pri potencah x ([2]). Temu vprašanju se lahko izognemo z omejitvijo stopnje Čebiševega polinoma. Če je interval majhen in ima funkcija malo ničel, bodo ničle polinoma dobra aproksimacija ničlam funkcije f. Manjše napake lahko hitro popravimo z enim ali dvema iteracijama, na primer z uporabo Newtonove iteracije na f. Če je interval velik in ima veliko ničel, pa se bomo morda morali poslužiti delitvi intervala na podintervale in postopek iskanja uporabiti za vsak podinterval posebej. Z zavedanjem vseh teh omejitev lahko z algoritmom CTP pridemo do ničel s strojno natančnostjo. Prednost algoritma je tudi, da ne potrebujemo nobenega predznanja o ničlah.

3.1.1 Izračun Čebiševih koeficientov

Naša Čebiševa aproksimacija je končna vrsta Čebiševih polinomov, ki interpolirajo f na množici $(N + 1)$ točk, predstavljenih v (3.1), znanih kot Čebišev-Lobattova mreža. Z množenjem kvadratne matrike z vektorjem, ki vsebuje vrednosti funkcije f v izbranih točkah, dobimo Čebiševe koeficiente v obliki vektorja. Če je evaluiranje f drago, je najbolje, da omejimo N na potence števila 2. V tem primeru, ko N povečamo, lahko uporabimo vse že prej izračunane vrednosti in tako največje število evaluacij nikoli ne preseže

najmanjšega N, ki ustreza zaustavitvenemu kriteriju. Čebiševa vrsta za analitično funkcijo f na intervalu $[a, b]$ konvergira, saj sta j-ti izraz in s tem tudi absolutna vrednost j-tega koeficienta a_j omejena z ρ^j za nek $\rho < 1$. Napaka interpolacije na $(N + 1)$ točkah je tipično enakega reda kot zadnji izračunan Čebišev koeficient a_N. Po [2] obstajata dva zaustavitvena kriterija za N. Eden temelji na Čebiševem koeficientu in pravi, da povečujemo N, dokler ni

$$\sum_{j=[(2/3)N]}^{N} |a_j| < \varepsilon,$$

kjer $[(2/3)N]$ predstavlja celo število, ki je najbližje $2N/3$. Drugi kriterij pa temelji na evaluiranih točkah in pravi, da mora biti

$$\max_j |f_{2N}(x_j) - f_N(x_j)| < \varepsilon,$$

kjer je ta razlika izračunana za vse točke na mreži $(2N + 1)$ točk, ki niso vsebovane na bolj grobi mreži $(N+1)$ točk. V točkah, skupnim obema mrežama, sta vrednosti interpolantov seveda enaki in kriterij nima pomena. Ker napaka aproksimacije f_N teži k maksimumu v okolici sredine med točkama aproksimacije, je napaka v teh sredinskih točkah zelo verjetno zelo blizu pravemu maksimumu napake po točkah.

3.1.2 Delitev intervalov

Kot bomo bolj natančno spoznali v 3.1.5, je metoda CTP dobro definiran proces le pod pogojem, da je N omejen na neko zmerno stopnjo. V primeru, da za pridobitev dovolj natančne aproksimacije potrebujemo velik N, pa se moramo poslužiti delitve intervala. Priporoča se, da vseeno aproksimiramo f na celotnem intervalu, četudi zahteva velik N. Če najvišjo stopnjo, ki še prinese zadovoljive rezultate pri CTP označimo z N_{max}, potem je priporočljivo interval razdeliti na $[N/N_{max}]$ podintervalov, kjer $[N/N_{max}]$ predstavlja najbližje celo število. Ko zaključimo postopek delitve, algoritem nato uporabimo na vsakem podintervalu posebej brez dodatnih sprememb. Več o delitvi intervalov si bomo pogledali tudi v 3.3.

3.1.3 Skaliranje

Čebiševi razvoji so zelo enakomerni v smislu, da maksimalna napaka, gledano po točkah, oscilira z špicami in koriti z enako amplitudo po celotnem intervalu $[a, b]$. Če je f že sama po sebi zelo neenakomerna, kot na primer funkcija $f(x) = \exp(10x)\sin(x)$, potem bo Čebiševa vrsta imela velike relativne napake, kjer je f zelo majhna. V takih primerih se lahko poslužujemo dveh sredstev ([2]). Kot prvo lahko interval razdelimo na podintervale, ki bodo tako majhni, da se f kar najmanj razlikuje na vsakem podintervalu. Druga možnost pa je, da f pomnožimo z gladko funkcijo, ki bo zgladila velike skoke pri f. Če pogledamo skozi prejšnji primer, ima $\tilde{f}(x) = \exp(-10x)f(x) = \sin(x)$ enake ničle kot f, ker pa je veliko bolj enakomerna, bo imel Čebišev razvoj $\tilde{f}$ veliko manjšo relativno napako in bo dal veliko natančnejši približek za ničle. Preoblikovanje gladke funkcije za skaliranje pa ni lahko delo.

3.1.4 Negladke funkcije

Čebiševi razvoji pri negladkih funkcijah konvergirajo, ampak zelo počasi. Če ima f pole, točke nezveznosti ali druge singularnosti na razvojnem intervalu $[a, b]$, vključno s singularnostmi v krajiščih intervala, ugotovimo, da algoritmi ne delujejo. Na žalost mora biti funkcija analitična v vseh točkah izbranega intervala, vključno z mejnima. Če funkcija vsebuje kakršnokoli od točk nezveznosti oz. drugih singularnosti izven intervala, pa to na algoritem seveda ne vpliva.

3.1.5 Od končne Čebiševe vrste do polinoma

Interval $[a, b]$ z uporabo translacije (3.2) preslikamo na interval $[-1, 1]$. Čebišev razvoj funkcije f na $[a, b]$ je tako

$$f_N(x) = \sum_{j=0}^{N} a_j T_j(y) = \sum_{j=0}^{N} b_j y^j, \quad y = y(x) \in [-1, 1].$$

Zanima nas, kako učinkovito izračunati koeficiente $(b_j)_j$ iz danih koeficientov $(a_j)_j$. Ceno transformacije lahko razpolovimo z delitvijo f_N na sode in lihe dele. Naj bosta $\mathbf{a}^{sod}$ in $\mathbf{b}^{sod}$ vektorja, ki vsebujeta koeficiente pri sodih stopnjah, to je $a^{sod} = \{a_0, a_2, a_4, \dots\}$ in $b^{sod} = \{b_0, b_2, b_4, \dots\}$. Potem je

$$\mathbf{b}^{sod} = \mathbf{Q}^{sod}\mathbf{a}^{sod}.$$

Elemente matrike $\mathbf{Q}^{sod}$ na diagonalah izračunamo po formulah, predstavljenih v [2],

$$q_{11}^{sod} = 1, \quad q_{jj}^{sod} = 2^{2j-3}, \quad j = 2, 3, \dots.$$

Za izračun ostalih elementov pa uporabimo naslednjo rekurzivno zvezo

$$q_{j-K,j}^{sod} = \left\lceil -\frac{(2j-2K)(2j-2K-1)}{2K(4j-2K-4)} q_{j-K+1,j}^{sod} \right\rceil,$$

pri čemer računamo elemente navzgor po posameznih stolpcih. Podobno dobimo elemente lihe matrike $\mathbf{Q}^{lih}$ po formulah

$$q_{jj}^{lih} = 2^{2j-2}, \quad j = 1, 2, 3, \dots$$

in

$$q_{j-K,j}^{lih} = \left\lceil -\frac{(j-K)(2j-2K+1)}{K(4j-2K-2)} q_{j-K+1,j}^{lih} \right\rceil.$$

Kot primer navedimo

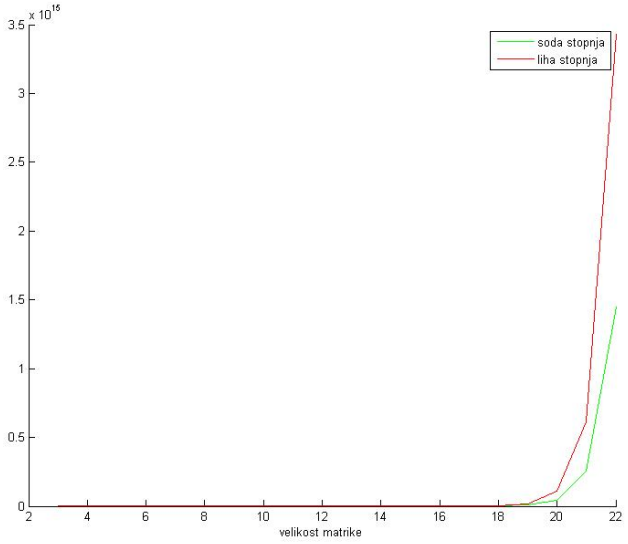
$$\mathbf{Q}^{sod} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 2 & -8 & 18 & -32 & 50 & -72 & 98 & -128 \\ 0 & 0 & 8 & -48 & 160 & -400 & 840 & -1568 & 2688 \\ 0 & 0 & 0 & 32 & -256 & 1120 & -3584 & 9408 & -21504 \\ 0 & 0 & 0 & 0 & 128 & -1280 & 6912 & -26880 & 84480 \\ 0 & 0 & 0 & 0 & 0 & 512 & -6144 & 39424 & -180224 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2048 & -28672 & 212992 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8192 & -131072 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32768 \end{bmatrix}.$$

in

$$\mathbf{Q}^{lih} = \begin{bmatrix} 1 & -3 & 5 & -7 & 9 & -11 & 13 & -15 & 17 \\ 0 & 4 & -20 & 56 & -120 & 220 & -364 & 560 & -816 \\ 0 & 0 & 16 & -112 & 432 & -1232 & 2912 & -6048 & 11424 \\ 0 & 0 & 0 & 64 & -576 & 2816 & -9984 & 28800 & -71808 \\ 0 & 0 & 0 & 0 & 256 & -2186 & 16640 & -70400 & 239360 \\ 0 & 0 & 0 & 0 & 0 & 1024 & -13312 & 92160 & -452608 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4096 & -61440 & 487424 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16384 & -278528 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 65536 \end{bmatrix}.$$

S tem, ko smo števila pri rekurziji zaokrožili na cela števila, smo se znebili napak pri zaokroževanju. Slika 3.2 prikazuje, kako skoraj eksponentno raste norma matrik, približno z

$$||\mathbf{Q}^{sod}||_\infty \sim 0.016(5.8)^j, \quad ||\mathbf{Q}^{lih}||_\infty \sim 0.039(5.8)^j.$$



Slika 3.2: Neskončne norme matrik $\mathbf{Q}^{sod}$ in $\mathbf{Q}^{lih}$ v odvisnosti od velikosti matrike.

Če se želimo izogniti žrtvovanju več kot 6 decimalnih mest natančnosti, oziroma, če želimo zagotoviti, da so napake pri koeficientih v potenčni obliki manjše od milijon-kratnika plavajoče vejice in zaokrožitvenih napak pri Čebiševih koeficientih, se moramo pri velikosti $\mathbf{Q}^{sod}$ in $\mathbf{Q}^{lih}$ omejiti na 9 ali manj, saj je pri matrikah velikosti 9×9

$$||\mathbf{Q}^{sod}||_\infty = 243712 \quad \text{in} \quad ||\mathbf{Q}^{lih}||_\infty = 559104.$$

Ko združimo sode in lihe polinome, dobimo nesimetričen polinom stopnje 17.

3.1.6 Metoda DD

Alternativno metodo za pridobitev polinoma v standardni bazi iz zapisa polinoma po Čebiševi bazi dobimo iz naslednjega izreka.

Izrek 3.1 ***Pridružen polinom podvojene stopnje***
Naj bo f_N dan polinom,

$$f_N(x) = \sum_{j=0}^{N} a_j T_j(x).$$

Definirjamo polinom h s podvojeno stopnjo kot

$$h_{2N}(z) = \sum_{j=0}^{2N} b_j z^j,$$

kjer so

$$b_j = \begin{cases} a_{j-N}, & j > N \\ 2a_0, & j = N \\ a_{N-j}, & j < N \end{cases}.$$

Potem so ničle x_k od f_N na realnem intervalu $[-1, 1]$ povezane z ničlami z_k polinoma h_{2N} na enotski krožnici v kompleksni ravnini preko relacije

$$x_k = \Re(z_k).$$

Dokaz. Iz identitete (2.3), to je $T_j(x) = \cos(jt)$ za $x = \cos(t)$ in $\cos(t) = (\exp(it) + \exp(-it))/2$, dobimo

$$f_N(\cos(t)) = \sum_{j=0}^{N} a_j \left(\exp(it) + \exp(-it)\right)/2.$$

Definirajmo $z = \exp(it)$ in

$$h_{2N}(z) = 2\exp(iNt) f_N(\cos(t)).$$

Ker za $\exp(iNt)$ velja, da ni nikoli nič, so ničle produkta enake tistim od $f_N(\cos(t))$. Uporaba $\exp(ijt) = e^{ijt} = (e^{it})^j = [\exp(it)]^j$ zaključi dokaz. □

3.1.7 Računanje ničel na celotni realni osi

Če ima funkcija neskončno ničel, je seveda nemogoče vse določiti numerično. Kljub temu je pogosto možno poiskati asimptotično aproksimacijo za ničle pri velikem $|x|$ in nato numerično izračunati končno število ničel, pri katerih je $|x|$ premajhen, da bi bila asimptotična formula natančna. Če vzamemo za primer Besselovo funkcijo J_0, k-to ničlo, označeno z $j_{0,k}$, asimptotično aproksimiramo z ([2])

$$j_{0,k} \sim \left(k - \frac{1}{4}\pi\right) + \frac{1}{8(k-\frac{1}{4})\pi} - \frac{31}{6(4k-1)^3\pi^3} + O(k^{-5}).$$

Ta aproksimacija ima absolutno napako 0.0018 za prvo ničlo ter napako 2.7×10^{-10} za dvajseto.

Če ima f le končno število ničel na neomejenem intervalu, je mogoče vse ničle poiskati direktno z uporabo zamenjave koordinat, ki nam interval preslika na kanonični interval za Čebiševe polinome $[-1, 1]$. Po tem lahko uporabimo kateregakoli od predhodno obravnavanih algoritmov na tem intervalu brez dodatnih sprememb. Če označimo koordinato na neskončnem intervalu z y, potem je lahko ena od možnosti za zamenjavo koordinat

$$y = \frac{Lx}{\sqrt{1-x^2}}, \quad x = \frac{y}{\sqrt{L^2+y^2}}, \quad x \in [-1, 1], y \in [-\infty, \infty],$$

kjer z L označimo konstanto, ki si jo izberemo glede na preslikavo. Četudi je optimalna izbira L odvisna od problema, Čebiševa konvergenca ni zelo občutljiva na njeno izbiro in se pogosto že izbira $L = 1$ izkaže kot dobra pri večini problemov.

Kot drugi primer lahko pogledamo funkcijo

$$f(y) = (2y^2 - 1) \exp\left(-\frac{1}{2}y^2\right),$$

ki ima le dve ničli na neskončnem intervalu, to sta $y_* = \pm 1/\sqrt{2}$. Po preslikavi postane

$$f(y) = \left(2\frac{L^2x^2}{1-x^2} - 1\right) \exp\left(-\frac{1}{2}\frac{L^2x^2}{1-x^2}\right).$$

Ker je postopek simetričem glede na izhodišče, ga lahko izboljšamo in pohitrimo in funkcijo f razširimo v Čebiševo vrsto samo na intervalu $[0, 1]$. Ta funkcija ima samo eno končno ničlo na celotni pozitivni realni osi v $y = 1/\sqrt{2}$, kar je ekvivalentno $x = 1/\sqrt{3}$ v transformirani koordinati.

3.2 Čebišev-Frobeniusova matrika

V do sedaj napisanem smo se srečali s problemom, kako iz gladke funkcije f izračunati končno Čebiševo vrsto in nato iz nje pridobiti polinom v standardni bazi. Sedaj pa bomo prikazali, kako iz polinoma v Čebiševi bazi izračunati ničle. Več kot stoletje nazaj je Georg Frobenius pokazal ([1]), da so ničle polinoma v obliki

$$f_N(x) = \sum_{j=0}^{N} b_j x^j \tag{3.3}$$

enake lastnim vrednostim matrike, ki se imenuje pridružena matrika polinoma in je enaka

$$B_{f_N} = (b_{jk})_{j,k=1}^{N}, \quad b_{jk} = \begin{cases} \delta_{j,k-1}, & j = 1, 2, \ldots, (N-1) \\ (-1)\frac{b_{j-1}}{b_N}, & j = N \end{cases},$$

kjer je δ Kronecherjev delta, za katerega velja

$$\delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}.$$

Pridružena matrika polinoma stopnje $N = 5$ je tako oblike

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{b_0}{b_5} & -\frac{b_1}{b_5} & -\frac{b_2}{b_5} & -\frac{b_3}{b_5} & -\frac{b_4}{b_5} \end{bmatrix}.$$

Obstajajo številne metode za računanje ničel polinoma, od Laguerrove metode do Jenkins-Traub argoritma. Kljub temu je iskanje lastnih vrednosti pridružene Frobeniusove matrike precej učinkovito. Kot zanimivost lahko omenimo, da je v Matlabu ta metoda uporabljena kot privzeta. Največja slabost te metode pa je, da je za izračun vseh ničel polinoma stopnje N potrebnih $10N^3$ operacij v plavajoči vejici.

Poznamo dve strategiji uporabe Frobeniusove matrike oz. iskalnikov ničel, ki temeljijo na potenčnih koeficientih b_j kakršnega koli tipa. Čebiševe koeficiente $\{a_j\}_j$ lahko pretvorimo v potenčne koeficiente $\{b_j\}_j$ z množenjem vektorja in matrike, kot je to opisano v poglavju 3.1.5. Kljub temu napake rezultata, glede na napake podatkov, rastejo z 2.4^N, tako da lahko to metodo varno uporabimo le, kadar je $N < 17$.

K sreči obstaja metoda za generiranje Frobeniusove matrike za katerikoli nabor ortagonalnih polinomov ([1]). Za splošen N so elementi Čebišev-Frobeniusove matrike enaki

$$a_{jk} = \begin{cases} \delta_{2,k}, & j = 1, k = 1, 2, \ldots, N \\ \frac{1}{2}\left(\delta_{j,k+1} + \delta_{j,k-1}\right), & j = 2, \ldots, (N-1), k = 1, 2, \ldots, N \\ (-1)\frac{a_{j-1}}{2a_N} + (1/2)\delta_{k,N-1}, & j = N, k = 1, 2, \ldots, N \end{cases}.$$

Za primer $N = 5$ Čebišev-Frobeniusova matrika izgleda

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ -\frac{a_0}{2a_5} & -\frac{a_1}{2a_5} & -\frac{a_2}{2a_5} & -\frac{a_3}{2a_5}+1/2 & -\frac{a_4}{2a_5} \end{bmatrix}.$$

Numerične raziskave kažejo ([1]), da je metoda dobra in natančna tudi v bližini večkratnih ničel.
Metode za delitev intervalov, ki jih bomo spoznali v nadaljevanju, morajo kot končni korak najti ničle polinomov nizke stopnje M na vsakem podintervalu, kjer je $M \ll N$. Pri ocenah časovne zahtevnosti, ki jih bomo predstavili v nadaljevanju, bomo upoštevali, da je uporabljena Čebišev-Frobeniusova metoda za iskanje lastnih vrednosti za ta končni korak s ceno okrog $10M^3$ operacij za vsak podinterval. S ceno bomo tako v nadaljevanju poimenovali število računskih operacij.

3.3 Algoritmi z delitvami intervalov

Polinom f_N stopnje N v obliki (3.3) lahko vedno predstavimo v bazi Čebiševih polinomov. To nam, kot bomo spoznali v nadaljevanju, omogoča boljše pogoje, kot če bi računali z originalnim polinomom v standardni bazi. Obstaja veliko metod za iskanje ničel, ki uporabljajo delitev intervalov. Algoritmi, ki jih bomo spoznali, razdelijo interval $[-1, 1]$ na N_S podintervalov in aproksimirajo f_N s Čebiševim interpolantom majhne stopnje na vsakem izmed podintervalov. V nadaljevanju bomo pokazali, da je ta metoda veliko cenejša za višje stopnje N.

3.3.1 Napaka delitve

Algoritmi za delitev intervala, ki jih bomo opisali v 3.3.2, temeljijo na delitvi intervala in nato uporabi različnih Čebiševih aproksimacij z različno stopnjo M na vsakem podintervalu. Da bi te algoritme lahko uporabljali,

moramo vedeti, kako izbrati primerno stopnjo M in velikost podintervala, da bo izbrana aproksimacija "prvotnega" polinoma f_N, kjer je seveda $N \gg M$, sprejemljiva. Odgovor na ti dve vprašanji bomo predstavili v tem razdelku.
Pri izbiri stopnje M lokalne aproksimacije se moramo zavedati, da je aproksimacija polinoma T_N najtežji problem pri aproksimaciji po Čebiševi bazi razvitega f_N, saj polinomi nižje stopnje oscilirajo počasneje. Zato potrebujemo izrek, ki nam bo povedal, kako natančno je T_N aproksimirana s shemo N_S podintervalov in lokalnih polinomov stopnje M.
Opazimo tudi, da T_N oscilira zelo hitro blizu $x = \pm 1$ in zelo počasi blizu $x = 0$. Kljub temu, z že omenjeno identiteto v (2.3), lahko ta odstopanja odstranimo s spremembo koordinate $x = \cos(t)$, ki končno Čebiševo vrsto spremeni v končno cosinusno vrsto oz. trigonometrično vrsto z istimi koeficienti:

$$f_N^{trig}(t) = f_N(\cos(t)) = \sum_{j=0}^{N} a_j \cos(jt). \tag{3.4}$$

Kosinus v trigonometrični vrsti se bo obnašal enako na vsakem podintervalu, če bodo delitve enakomerne. Dovolj je, da proučimo napako aproksimacije na intervalu $[0, \pi/N_S]$, saj napaka ne bo slabša in ne boljša na vseh ostalih podintervalih.
Vsaka lokalna Čebiševa vrsta bo na intervalu $\left[(k-1)\frac{\pi}{N_S}, k\frac{\pi}{N_S}\right]$, $k = 1, 2, \ldots, N_S$ uporabljala transformirano koordinato

$$z_k = \frac{2N_S}{\pi}\left(t - \frac{\pi}{N_S}\left(k - \frac{1}{2}\right)\right).$$

Za dokaz želenega izreka razvijemo

$$\cos(Nt) = \cos\left(N\frac{\pi}{2N_S}\left(z_k + 2k - 1\right)\right).$$

Z uporabo trigonometričnih identitet lahko to zapišemo kot linearno kombinacijo

$$\sin(N\frac{\pi}{2N_S}z_k) \quad \text{in} \quad \cos(N\frac{\pi}{2N_S}z_k).$$

Znan izrek o napaki interpolacijskega polinoma pove, da je napaka pri interpolaciji funkcije g s polinomom z $(M+1)$ interpolacijskimi točkami z_j

enaka

$$|g(z) - g_M(z)| = \frac{1}{(M+1)!}\left|g^{(M+1)}(\xi)\right|\left|\prod_{j=0}^{M}(z - z_j)\right| \tag{3.5}$$

za neko vrednost ξ, ki leži na intervalu, ki ga razprostira z in interpolacijske točke z_j. V primeru, ko so interpolacijske točke enake ničlam T_N, velja po izreku 2.1, da je

$$\max_{z\in[-1,1]} \prod_{j=0}^{M}(z - z_j) = \frac{1}{2^M}.$$

Iz tega je razvidno, da je

$$\max_{z\in[-1,1]} |g(z) - g_M(z)| \leq \frac{1}{(M+1)!} \max_{z\in[-1,1]} \left|g^{(M+1)}(z)\right| \frac{1}{2^M}.$$

Definirajmo $Q := N_S/N$. Za določeno funkcijo

$$g(z) = \cos\Big(\frac{\pi}{2Q}z + faza\Big),$$

ki je najbolj hitro oscilirajoča komponenta $f_N^{trig}(t(z))$, ko je uporabljenih $N_S = NQ$ podintervalov, lahko določimo mejo, neodvisno od faze,

$$\max_{z\in[-1,1]} |g^{(M+1)}(x)| \leq \frac{\pi^{M+1}}{2^{M+1}Q^{M+1}}.$$

Ko vstavimo to mejo v neenakost (3.5), dobimo sledeči izrek.

Izrek 3.2 ***Izrek o lokalni aproksimaciji*** $\cos(Nt)$
Imamo polinom f_N stopnje N s Čebiševimi koeficienti a_j. Predpostavimo, da so koeficienti normirani, $\sum_{j=0}^{N} |a_j| = 1$, kar lahko dosežemo z deljenjem prvotnega polinoma z vsoto absolutnih vrednosti njegovih Čebiševih koeficientov, ne da bi spremenili ničle. S tem dosežemo, da je $\max|f_N(t)| \leq 1$ *za vse $t \in [0, \pi]$. Interval $[0, \pi]$ razdelimo na N_S delov. Interpoliramo $f_N^{trig}(t) = f_N(\cos(t))$ na k-tem podintervalu s Čebiševo vrsto stopnje M v t:*

$$f_N^{trig}(t) \approx F_k(t; M) = \sum_{j=0}^{M} c_j^k T_j\left(\frac{2N_S}{\pi}\left(t - \frac{(2k-1)\pi}{2N_S}\right)\right), \quad t \in \frac{\pi}{N_S}[k-1, k].$$

Sledi, da je za vsak k napaka pri aproksimaciji f_N z lokalnimi interpolanti na vsaki delitvi intervala $[0, \pi]$ v trigonometrični koordinati t omejena z

$$\max |f_N(t) - F_k(t; M)| \leq \frac{\pi^{M+1}}{(M+1)!2^{2M}Q^{M+1}},$$

pri čemer so interpolacijske točke ničle Čebiševega polinoma stopnje $M + 1$.

Da dosežemo zmerno napako aproksimacije, na primer 10^{-5}, je postopek dokaj enostaven. Za N podintervalov zadošča že aproksimacija s polinomom sedme stopnje, medtem ko za $2N$ podintervalov minimalen M pade na $M = 6$. Za polinom pete stopnje pa zadostuje pogoj $Q = 3$. Da pa dosežemo zelo nizke napake, kot na primer 10^{-15}, potrebujemo bodisi N polinomov stopnje 12, $2N$ lokalnih aproksimacij stopnje 9, $10N$ aproksimacij s polinomom stopnje 8 ali pa $25N$ aproksimacij s polinomom stopnje 5 ([1]).

3.3.2 Algoritmi za delitev intervala

Osnovna ideja metod delitve intervalov je, da aproksimiramo polinom f_N na določenem podintervalu s Čebiševimi polinomi neke majhne stopnje M in nato poiščemo ničle vsake lokalne aproksimacije z uporabo Čebišev-Frobeniusove metode oz., če je $M \leq 3$, po znanih eksplicitnih formulah za ničle kubične, kvadratne oz. linearne funkcije. Ničle polinoma f_N na intervalu $[-1, 1]$ so nato unija vseh sprejemljivih ničel lokalnih aproksimacij, to je ničel, ki so realne in ležijo na intervalu $[-1, 1]$. Algoritmov ni težko sprogramirati. Cena Čebišev-Frobeniusovega koraka je $10M^3N_S$, kjer je N_S število delitev.
Dodatno pa moramo izračunati N_S Čebiševih interpolantov. Prvi in najdražji korak pri tem je evaluacija $f_N^{trig}(t)$ iz (3.4) v vsaki od $(M + 1)N_S$ interpolacijskih točk.
Ko imamo enakomerno razporejene točke, kot na primer pri algoritmu Megakubi, obravnavanem v 3.3.5, je uporaba hitre Fourierove transformacije (v nadaljevanju FFT) oz. bolj natančno diskretne kosinusne transformacije (v nadaljevanju DCT) najboljši način za izračun vrednosti od f_N^{trig}. Vendar pa je polinomska interpolacija z uporabo enakomerno razporejenih interpolacijskih točk v vsaki poddomeni smiselna le ob pogoju, da imamo majhen M,

kot bo to uporabljeno pri algoritmu, opisanem v 3.3.5. Interpolacija na enakomerno razporejenih točkah že pri uporabi zmernega M pripelje do težav, saj je zelo slabo pogojena in pogosto nekonvergentna. Zato smo prisiljeni uporabiti lokalne Čebiševe točke, pri čemer pa seveda globalne točke niso enakomerno razporejene. Ker lahko FFT uporabimo le pri tem pogoju, vseh potrebnih točk ni mogoče izračunati z uporabo le ene FFT.

Več o uporabi FFT in DCT za računanje točk si lahko preberemo v [1]. Ne glede na to, kako izračunamo vrednosti f_N v interpolacijskih točkah x_i, moramo v vsakem primeru pridobiti koeficiente lokalnega polinoma iz vrednosti v točkah. To naredimo z množenjem $(M+1)$ vrednosti na vsaki poddomeni z $(M+1)\times(M+1)$ matriko za izračun Čebiševih koeficientov lokalne aproksimacije. Ta drugi korak nas stane približno $2N_SM^2$, kar je običajno zanemarljivo v primerjavi z izračunom vrednosti $f_N(x_i)$. Kljub temu je skupna cena z uporabo parametra $Q = N_S/N$ za majhen M in eno samo FFT približno enaka ([1])

$$C^{FFT} \approx (5/2)QNM\left(4M^2 + \log_2(QNM)\right) \tag{3.6}$$

$$\approx (5/2)QNM\log_2(QNM), \quad \log_2(QNM) \gg 4M^2. \tag{3.7}$$

Kadar pa uporabimo zmeren ali velik M in $(M+1)$ hitrih FT, cena znaša

$$\begin{aligned} C^{\text{več } FFT} \approx & NQ(10M^3) + NQ(2M^2) \\ & + 5(M+1)N\max(Q,1)\log_2(2N\max(Q,1)). \end{aligned}$$

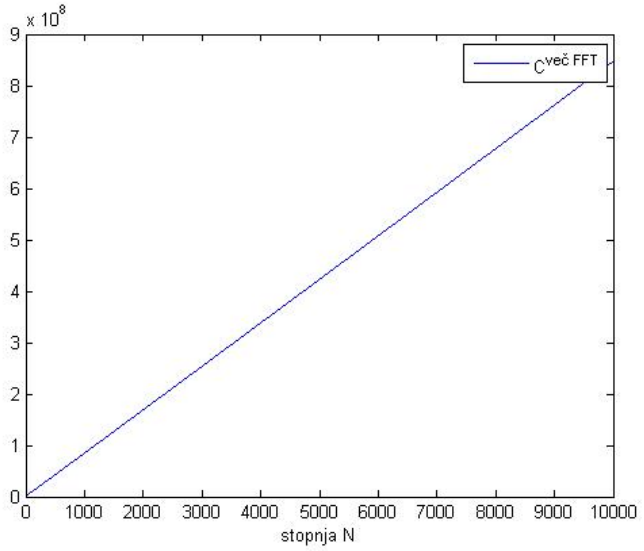
Ocene cen temeljijo na treh parametrih N, Q, M. Da bi bolje razumeli, zakaj so pomembni, moramo najprej določiti toleranco napake in nato uporabiti teorijo napake delitve, kar smo prikazali v 3.3.1. Da bi napaka lokalne aproksimacije na poddomeni padla pod izbrano toleranco ϵ, moramo izbrati Q tako, da je

$$Q(M) = \pi\frac{1}{(\epsilon(M+1)!2^{2M+1})^{1/(M+1)}}. \tag{3.8}$$

Sedaj je dokaj enostavno evaluirati cene glede na izbran ϵ za poljubno stopnjo M lokalne aproksimacije in stopnjo N funkcije f_N. V naslednjih razdelkih bomo analizirali ceno za dva algoritma, tistega, ki uporablja več FFT z zmernim M in tistega, ki uporablja en FFT z majhnim M.

3.3.3 Subdivizijski algoritem 13-N ("Tredecic")

Neprikladno bi bilo izbrati M, ki bi bil odvisen tako od N kot tudi od tolerance ϵ, ampak na srečo to ni potrebno. Recimo, da si postavimo visoko mejo tolerance za napako, na primer 10^{-12}. Kljub temu, da je meja kar visoka, to lahko koristi pri zagotovljanju majhne absolutne napake in s tem zadovoljivo majhne relativne napake. Izbira še nižje tolerance pa se odraža v ceni računanja, saj napake pri seštevanju Čebiševe vrste znašajo običajno $O(N)$-krat osnovna zaokrožitvena napaka, na primer 2×10^{-16} za računalnike, ki uporabljajo IEEE standard plavajoče vejice.
Če narišemo ceno $C^{\text{več } FFT}$ na grafu (M, N) z Q, izbranim s (3.8), se izkaže, da je cena skoraj linearna z N in da se ne razlikuje veliko z izbrano lokalno stopnjo M v okolici minimalne cene. Slika 3.3 prikazuje ceno v odvisnosti od stopnje N pri fiksni izbiri $M = 13$ in z Q, poračunanim po (3.8).



Slika 3.3: Graf cene $C^{\text{več } FFT}$ v odvisnosti od stopnje N pri fiksni izbiri $M = 13$.

Fiksna izbira $M = 13$ in $Q = 1$, kar pomeni, da je število podintervalov enako stopnji N, je zelo blizu minimuma za $50 \leq N \leq 10000$.

Ime "Tredecic" dobimo iz strategije uporabe fiksnega $M = 13$ neodvisno od N. Lokalne aproksimacije bodo na vsakem podintervalu stopnje 13. Ime 13-N je prav tako primerno, saj je število podintervalov enako N.
Cena algoritma je poseben primer $C^{\text{več } FFT}$ in je enaka

$$C^{Tredecic} \approx N\{22000 + 70\log_2(N)\} \approx 22000N \tag{3.9}$$

operacij v plavajoči vejici. Logaritmični del lahko zanemarimo, saj znaša manj kot 5% od 22000 za vse $N < 30000$. Algoritem 13-N je cenejši kot uporaba Čebišev-Frobenius-ovega algoritma na Čebiševi vrsti stopnje N, ko je $N \geq 50$.
Ceno lahko skoraj prepolovimo z izbiro napake 10^{-8} in izbiro $M = 10$ ter enako kot pri predhodno opisanem, številom podintervalov enakim stopnji N. Ta algoritem lahko poimenujemo "Dedic" in je cenejši od Čebišev-Frobeniusovega za vse $N > 33$.

3.3.4 Nižanje cene z določanjem "brezničelnih" intervalov

Polinom stopnje N ima največ N ničel, zato veliko lokalnih aproksimacij nima realnih ničel. Algoritme, ki uporabljajo delitev intervala, lahko zelo pohitrimo, če najdemo lahek in poceni način za določanje intervalov, ki ne vsebujejo nobene ničle. S tem se znebimo računanja ničel na vsakem brezničelnem intervalu in z vsakim prihranimo več kot 22000 operacij pri npr. 13-N algoritmu. Predstavimo izrek, ki prinese par pogojev za brezničelnost. Izrek se nanaša na splošen polinom stopnje N in ga je potrebno pri algoritmih, ki uporabljajo delitev intervala, uporabiti za vsak podinterval posebej. Interval $[-1, 1]$ moramo seveda zamenjati s k-tim podintervalom.

Izrek 3.3 ***(Brezničelni kriterij)***
Naj bo f_N polinom, zapisan v bazi Čebiševih polinomov:

$$f_N(x) = \sum_{j=0}^{N} a_j T_j(x), x \in [-1, 1].$$

Definirajmo

$$B_0 := \sum_{j=1}^{N} |a_j|, \quad B_1 := \sum_{j=1}^{N} |ja_j|,$$
$$h = \pi/N, \quad t_j = \pi j/h, j = 0, 1, \ldots, N.$$

Potem velja sledeče.
(1) Če je $B_0 < |a_0|$ *ali*
(2) Če je $B_1 h(1/2) < \min_j |f_N(\cos(t_j))|$,
potem f_N *nima ničle na intervalu* $x \in [-1, 1]$.

Dokaz. Ker je $T_j(x) \leq 1$ za vse $x \in [-1, 1]$, je B_0 zgornja meja za vse nekonstantne izraze v končni Čebiševi vrsti. Če je ta meja pod $|a_0|$, ki je konstanta v Čebiševi vrsti, potem je nemogoče, da bi kateri od ostalih delov pripeljal f_N do ničle na tem intervalu. S tem zaključimo dokaz prve trditve.
Vrednost B_1 je zgornja meja prvega odvoda trigonometričnega kosinusnega polinoma, ki ima enake koeficiente kot f_N, to je $f_N^{trig}(t) = \sum_{j=0}^{N} a_j \cos(jt)$. To mejo dobimo tako, da odvajamo kosinusni polinom po t in z opazovanjem, da je $\sin(jt)$ omejen z ena.
Točke $f_N^{trig}(t_j)$ so enake vrednostim $f_N(x)$ v interpolacijskih točkah, transformiranih iz x v t po $x = \cos(t)$, ki so bile uporabljene za izračun a_j. Funkcija $|f_N^{trig}(t)|$ se od magnitude $f_N^{trig}(t_j)$ ne more spustiti bolj strmo kot linija padca - B_1, sicer bi bili v nasprotju z mejo prvega odvoda f_N^{trig}. Če vzamemo interval dolžine h s centrom v t_j, dobimo

$$|f_N^{trig}(t)| \geq |f_N^{trig}(t_j)| - B_1 h/2, \quad \forall t \in [t_j - h/2, t_j + h/2].$$

Če je torej minimum magnitude pozitiven, potem interval nima ničel. Če ta pogoj vstavimo v vse $N + 1$ interpolante, dobimo drugo trditev. □

Dokaz je povzet po [1].

3.3.5 Algoritem Megakubi ("Megacubes")

Ideja 13-N algoritma je, da aproksimiramo f_N z uporabo polinomov zmerne stopnje $M = 13$ na zmernem številu podintervalov $N_S = N$. Algoritem Megakubi pa po drugi strani uporablja veliko bolj ekstremno filozofijo o uporabi polinomov zelo nizke stopnje, v tem primeru kubičnih, na velikem številu podintervalov ([1]). Pri tem obstajajo številne olajšave. Kot prvo, uporaba neenakomerno razporejenih Čebiševih interpolacijskih točk na vsakem podintervalu ni potrebna. Shema, ki uporabi obe končni točki in točki na 1/3 in 2/3 podintervala bo zadoščala za stabilen izračun kubičnega interpolanta. Drugič, ker je potrebno, da so točke na grafu, gledano globalno, enakomerno razporejene in ne grupirane v podmnožice, je potrebno izvesti le eno FFT z namenom pridobitve vseh vrednosti f_N, ki jih potrebujemo za interpolacijo na vsaki poddomeni. Dalje, uporabimo lahko DCT in ne splošne FFT, kot jo zahteva algoritem 13-N. In tretjič, polinomi lokalnih transformacij so nizke stopnje, s tem pa se znebimo matrik za računanje ničel in jih zamenjamo z eksplicitnimi formulami za ničle kubičnih polinomov.
Kot smo videli v 3.3.4, z iskanjem neničelnih podintervalov zelo zmanjšamo zahtevnost 13-N algoritma. Pri algoritmu Megakubi si, ker je le poseben primer 3.2, pomagamo s sledečim kriterijem ki ga navaja [1].

Izrek 3.4 ***Brezničelni kriterij, ki temelji na lokalni interpolaciji***
Naj f_N^{trig}, definiran v (3.4)*, predstavlja kosinusni polinom stopnje N, ki je normaliziran tako, da velja $\sum_{j=0}^{N} |a_j| = 1$. Dalje, naj bo k-ti podinterval enak $[\alpha_k, \beta_k]$, kjer*

$$\alpha_k = (\pi/N_S)(k-1), \quad \beta_k = (\pi/N_S)k.$$

Če velja
(1) $\operatorname{sign}(f_N^{trig}(\alpha_k)) = \operatorname{sign}(f_N^{trig}(\beta_k))$*, kjer* sign *predstavlja ali je vrednost pozitivna oziroma negativna, ali če*
(2) $\min(|f_N^{trig}(\alpha_k)|, |f_N^{trig}(\beta_k)|) > (\pi^2/8)N^2/N_S^2$,
potem f_N^{trig} nima ničle na k-tem intervalu.

Z uporabo tega kriterija je cena rešitve kubičnih enačb zanemarljiva v primerjavi s ceno Fourierove transformacije.
Za interval dolžine π/N_S s štirimi enakomerno razporejenimi točkami, med katerimi sta dve meji intervala, lahko na osnovi izreka o napaki interpolacijskega polinoma ugotovimo, da je absolutna napaka pri aproksimaciji $\cos(Nt)$ omejena z

$$B_{kub,er} = \frac{\pi^4}{1944}\frac{N^4}{N_S^4} = 0.0501\frac{N^4}{N_S^4}.$$

Iz izraza je razviden pogoj, da mora biti za dosego dobre lokalne kubične aproksimacije N_S zelo velik v primerjavi z N.
Da si zagotovimo mejo napake na okoli 10^{-12}, kot smo si izbrali pri 13-N algoritmu, je potrebno izbrati $Q = 500$, kar pomeni $N_S = 500N$. Ceno lahko nato dobimo z modifikacijo iz (3.6)

$$C^{Megakubi}(N_S = 500N) \approx 40000N + 4000N\log_2(N).$$

Modifikacija, ki smo jo uporabili, je, da pri vrednosti $M = 3$, najdemo ničle precej ceneje z uporabo ekplicitnih formul za ničle kubičnih polinomov v primerjavi z Čebišev-Frobeniusovo metodo. Z izbiro te meje za napako je število operacij algoritma Megakubi večje od števila operacij pri 13-N algoritmu za vse N. Vendar pa, predvsem zato, ker uporabljamo $M = 3$ in ker napaka temelji le na N_S na četrto potenco, lahko število podintervalov drastično zmanjšamo z uporabo manjše tolerance za napako. Če za primer vzamemo $\epsilon = 10^{-8}$, lahko Q zmanjšamo za desetkratnik in se cena posledično zelo zniža

$$C^{Megakubi}(N_S = 50N) \approx 2700N + 400N\log_2(N).$$

Za enako mejo napake lahko algoritem 13-N zamenjamo z 10-N, kar pomeni $N_S = N$ s stopnjo lokalne aproksimacije 10. S tem cena pade za faktor 2 in posledično je 10-N algoritem dražji kot algoritem Megakubi z $N_S = 50N$.
Seveda lahko v vsak algoritem vpeljemo spremembe in razne izboljšave v delitve intervala. Kljub temu pa sta algoritma 13-N in Megakubi predstavnika opcij malo delitev z zmerno stopnjo lokalnih aproksimacij oz. veliko delitev z lokalnimi aproksimacijami nizke stopnje.

Kljub temu, da je Megakubi algoritem dražji od $N_S = N$ algoritmov pri visoki stopnji tolerance za napako, pa je ceneje spreminjati stopnjo tolerance za napako in je algoritem lažje sprogramirati. Ime algoritma izvira iz uporabe kubičnih interpolantov ter dejstva, da bo velikostni razred števila podintervalov milijon, če bo N velikosti tisoč.

Poglavje 4

Primeri uporabe

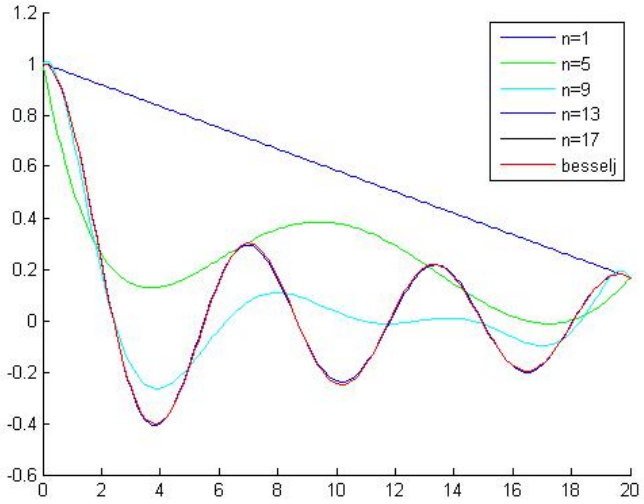
Kot prvi primer si poglejmo aproksimacije izbrane funlcije z uporabo interpolacije na Čebiševih točkah. Kot primer vzemimo Besselovo funkcijo prve vrste, ki je vgrajena v Matlab pod ukazom `besselj(v,z)`, kar predstavlja

$$J_v(z) = \left(\frac{z}{2}\right)^v \sum_{k=0}^{\infty} \frac{\left(\frac{-z^2}{4}\right)^k}{k!\Gamma(v+k+1)},$$

kjer je Γ gama funkcija,

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt.$$

Najprej bomo uporabili v programih priloženo funkcijo `cheb_interpolation( a,b,f,n )`, ki podano funkcijo f interpolira z uporabo n točk na intervalu $[a, b]$ in kot rezultat vrne anonimno funkcijo g in seznam koeficientov Čebiševega polinoma.



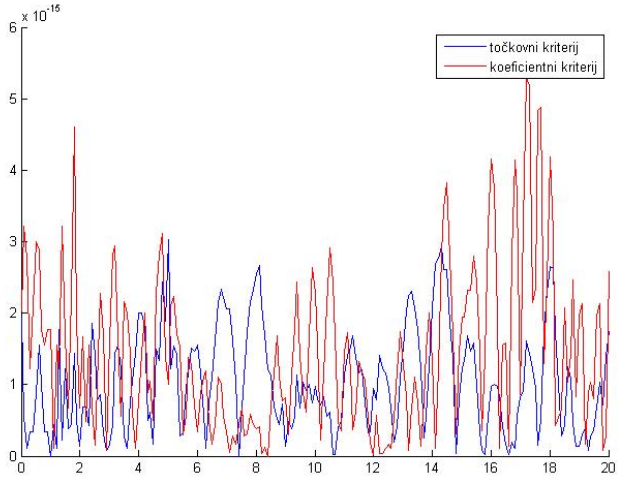
Slika 4.1: Graf interpolacije Besselove funkcije z uporabo 1, 5, 9, 13 in 17 interpolacijskih točk.

Na sliki 4.1 vidimo, kako se interpolacijska funkcija, ki jo dobimo s pomočjo Čebiševih polinomov, vedno bolj približuje podani funkciji in je napaka tako vedno manjša. Graf in interpolacijske funkcije so pridobljene s priloženo skripto `test_beseljInterpolation.m`.

Sedaj si lahko pogledamo primer uporabe interpolacije z uporabo zaustavitvenih kriterijev, obravnavanih v 3.1.1. V ta namen smo napisali funkciji `[ g,n ] = cheb_interpolation_koeff_criterion(a,b,f,e)` in `[ g,n ] = cheb_interpolation_point_criterion(a,b,f,e)`. Obe funkciji vrneta interpolacijsko funkcijo g stopnje n funkcije f na intervalu $[a, b]$ z napako manjšo od e. Za primer si vzemimo $e = 10^{-12}$.

Na sliki 4.2 vidimo primer interpolacije z uporabo dveh kriterijev. Opazimo lahko, da se aproksimaciji zelo dobro prilegata originalni funkciji, saj so napake zelo majhne. Kot zanimivost lahko povemo, da je aproksimacija, ki uporablja kriterij po točkah, potrebovala manj korakov in je aproksimirala z uporabo 32 interpolacijskih točk, medtem ko je funkcija, ki uporablja kriterij po koeficientih, aproksimirala z uporabo 64 interpolacijskih točk. Graf je

pridobljen s priloženo skripto `test_beseljCriterions.m`.



Slika 4.2: Graf napake pri interpolaciji Besselove funkcije z uporabo dveh kriterijev.

Sedaj pa si poglejmo še metode za iskanje ničel. Najprej si poglejmo metodo, ki smo jo obravnavali v 3.2. Za primer polinoma si vzemimo $p(x) = x^6 - 0.5$ in poiščimo njegove ničle z uporabo
(`[ A,r ] = chebFrobenious( a,b,p,n )`) , ki poišče ničle polinoma p na intervalu $[a, b]$ z uporabo n interpolacijskih točk in vrne Čebišev-Frobeniusovo matriko A ter ničle r:

```
p=@(x)x^6-0.5;
[ A,r ] = chebFrobenius( -1,1,p,6 );
```

Rešitev sta ničli

```
r =

  -0.890898718140338
   0.890898718140339.
```

Če uporabimo Matlabov algoritem `roots` za iskanje ničel polinoma, pa dobimo rezultat

ans =

$$\begin{array}{r} -0.890898718140340 \\ 0.890898718140341. \end{array}$$

Če izračunamo absolutno napako pri izračunu posamezne ničle, dobimo

$$1.665334536937735e^{-15}$$

za prvo in

$$2.220446049250313e^{-15}$$

za drugo ničlo. Primer je prikazan v priloženi skripti `test_chebFrobenius.m`.

Sedaj lahko pokažemo še delovanje algoritmov, ki uporabljajo subdivizijo intervalov in so obravnavani v 3.3.3 in 3.3.5. Priložene so funkcije
`megacubes(a,b,p,N,Q)`,`tredecic(a,b,p,N)` in `decic(a,b,p,N)`,
ki kot vhod zahtevajo interval $[a, b]$, polinom p in željeno število podintervalov N. Funkcija `megacubes` zahteva še dodaten parameter Q, kot je to opisano v poglavju 3.3.5. Kot rezultat nam funkcije vračajo
`[ g_list, sub_intervals, ai_list ]`,
kar predstavlja anonimno funkcijo na vsakem podintervalu, podintervale in koeficiente Čebiševega interpolacijskega polinoma na vsakem podintervalu. Le-te nato vstavimo v funkcijo za iskanje ničel, ki uporablja Čebišev-Frobeniusovo matriko, prilagojeno na poljuben interval $[a, b]$. Funkcijo kličemo z ukazom (`chebFrobenius_withChebKoeff( a,b,a_i )`).
Kot rezultat nam funkcija vrne Čebišev-Frobeniusovo matriko A in ničle r. V priloženih skriptah `test_tredecic.m`, `test_decic.m` in `test_megacubes.m` smo za primer iskali ničle polinoma

$$p = x^{63} - x^{11} - 0.1$$

na intervalu $[-1, 2]$ in spremljali čas računanja in natančnost pri računanju ničel v primerjavi z vgrajenim algoritmom `roots`. Le-ta nam vrne ničle

$$-0.997926579317150, \quad -0.811132212663578$$

in

$$1.001800227292726.$$

Največ časa nam vzame računanje ničel z algoritmom Megakubi s priloženo skripto `test_megacubes.m` in uporabljenim $Q = 500$. Čas računanja znaša približno 5.604738 sekund, natančnost ničel pa znaša

$$2.583605693433633e^{-04}, \quad 4.871551529683771e^{-05}$$

in

$$1.254817616724857e^{-04}.$$

Občutno hitrejši je algoritem 13-N, uporabljen v skripti `test_tredecic.m`, ki z uporabo $N = 20$ porabi le 0.232075 sekunde in najde ničle z natančnostjo

$$2.425792899884982e^{-11}, \quad 6.258993323626783e^{-12}$$

in

$$5.795298174682273e^{-08}.$$

To predstavlja občutno razliko tako v času, kot pri natančnosti. Tudi algoritem 10-N se izkaže dobro, saj v času 0.028258 sekunde izračuna ničle z natančnostjo

$$6.575317446078088e^{-08}, \quad 1.065149857204517e^{-09}$$

in

$$1.165089842380951e^{-07}.$$

Algoritem Megakubi bi z uporabo manjšega Q zelo pohitrili, npr. pri uporabi $Q = 2$ porabimo samo 0.022912 sekunde, vendar pa pri tem pridelamp večjo napako pri izračunu ničel, to je

$$0.070634035072619, \quad 8.627046446731956e^{-04}$$

in

$$0.034003064627858.$$

Za izboljšavo napisanih algoritmov vidimo, da bi ob upoštevanju brezničelnih kriterijev, obravnavanih v 3.3.3 in 3.3.5, algoritme pohitrili. Kot nadaljna izboljšava pa bi bila uporabna metoda, ki išče ničle direktno z uporabo Čebiševih polinomov in bi tako nadomestila uporabo polinomskih algoritmov ter bi, pri uporabi Čebiševih polinomov za aproksimacijo, iskanje ničel pohitrila.

Literatura

[1] J. P. Boyd, "Computing real roots of a polynomial in Chebyshev series form through subdivision", *Applied Numerical Mathematics*, št. 56, str. 1077-1091, 2006.

[2] J. P. Boyd, "Computing zeros on a real interval through Chebyshev expansion and polynomial rootfinding", *SIAM Journal on Numerical Analysis*, št. 40, zv. 5, str. 1666-1682, 2002.

[3] M. Hladnik, *ANALIZA 1*, Zapiski predavanj, Ljubljana, 2012.

[4] J. Kozak, *Numerična analiza*, Ljubljana, 2008.

[5] J. D. Cook, "Chebyshev Polynomials", neobjavljen, Februar 2008, dostopen na http://www.johndcook.com/ChebyshevPolynomials.pdf (dostop 17-01-2016).

[6] J. Heinonen, "Lectures on Lipschitz analysis", neobjavljen, dostopen na http://www.math.jyu.fi/research/reports/rep100.pdf (dostop 16-01-2016).

[7] http://www.chebfun.org (dostop 19-12-2015).

[8] http://en.wikipedia.org/wiki/Chebyshev_polynomials (dostop 17-01-2016).